BBN Systems and Technologies Corporation

A Subsidiary of Bolt Beranek and Newman Inc.

Report No. 7173

High-Throughput, Survivable Protocols for CDMA Packet-Radio Networks (SRNTN-74)

John Zavgren and Melisse Leib

March 1, 1990

Prepared By:

BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, MA 02138

Prepared for:

DARPA/ISTO
1400 Wilson Bl.
Arlington, VA. 22209

Sponsored by:

The Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

This research is supported by the Information Processing Technologies Office of the Defense Advanced Research Projects Agency under contracts: MDA-903-83-C-0173 & N00140-87-C-8910. The views and conclusions contained in this document are those of the authors and do not represent the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Army or the United States Government.

Report No. 7173

High-Throughput, Survivable Protocols for CDMA Packet-Radio Networks (SRNTN-74)

John Zavgren and Melisse Leib

March 1, 1990

Prepared By:

BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, MA 02138

Prepared for:

DARPA/ISTO
1400 Wilson Bl.
Arlington, VA. 22209

Sponsored by:

The Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

This research is supported by the Information Processing Technologies Office of the Defense Advanced Research Projects Agency under contracts: MDA-903-83-C-0173 & N00140-87-C-8910. The views and conclusions contained in this document are those of the authors and do not represent the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Army or the United States Government.

Contents

1. Introduction

The result of our research is a suite of practical, implementable algorithms that a software developer can use to build a high-throughput, survivable Code-Division Multiple Access (CDMA) mobile packet radio network with existing spread-spectrum packet-radio hardware. The Code Division channel increases network throughput by allowing simultaneous transmissions to different destinations on the same frequency band. We have conducted extensive simulations to verify the performance of these protocols and we have determined they produce a high-throughput, survivable network. These protocols are an evolutionary step in packet-radio network architecture that starts with earlier pioneering work, most notably the work of Jubin et al. that is the basis of SURAP [1,2,3]. We have adopted many SURAP algorithms to our CDMA network architecture, most notably its routing algorithms.

This work has been influenced to some extent by a particular application. Our CDMA algorithms are intended for the Low-cost Packet-Radio (LPR) hardware [4] that has several advanced features, most notably its power level and FEC rate can be adjusted on a packet-by-packet basis and it uses forward error correction (FEC) codes to compensate for bit errors. (The salient features of the LPR hardware are summarized in Appendix C.) Our algorithms can be implemented on other spread-spectrum packet radio devices but the network performance will be affected by the capabilities of the particular hardware. Nevertheless, the algorithms in this paper will produce a high-quality packet-radio network with any spread-spectrum packet radio device that can use multiple spreading codes.

Conventional spread-spectrum packet-radio networks such as the packet overlay for SINCGARS [5] and the PRnet with the SURAN Protocol (SURAP [2]), do not use code division; they broadcast their transmissions on a single common radio channel. This broadcast channel is an efficient way for packet radios to exchange routing information and it makes network entry simple, but unfortunately it has the undesirable side-effect of causing packet radios to receive packets that are not addressed to them. Because a packet radio cannot determine the identity of a received packet until it is processed, these "unwanted" receptions increase the packet radios' processing loads. And, because conventional packet radio hardware cannot receive more than one packet at a time, when transmissions overlap the first transmission can be received but all others are blocked until this reception concludes. Thus the broadcast channel increases the fraction of time that the packet radio nodes are in this blocking state and thereby increases the packet blocking rate and consumes channel bandwidth with retransmissions. A CDMA channel on the other hand has much lower blocking rates and thus higher throughput because the nodes can be selective about which packets they receive.

Although code division and code division multiple access (CDMA) are familiar concepts to communications researchers, to our knowledge a thorough CDMA network design has never been attempted before. In this paper we summarize the performance constraints inherent in both broadcast and CDMA multi-hop packet-radio networks and present the results of a study that compares the throughput of two protocol suites, the broadcast SURAP protocol suite and our new CDMA protocol suite. Our results

show that in general CDMA can significantly improve the throughput of a packet-radio network and in particular our CDMA protocols realize this performance improvement, provide many of the advantages of a broadcast channel and provide other benefits as well, such as effective congestion control and max/min fair end-to-end flow control.

The work in this paper presents results from a research project that investigated link-level and end-to-end flow control algorithms for CDMA packet radio networks. Our main objective is to document new link-level algorithms, but the performance properties of a packet-radio network are determined by all its constituent algorithms; the link-layer algorithms are only one component of a much larger algorithm suite. The need to establish a context for our performance results has compelled us to broaden the focus of our paper to include some non-link-level aspects of CDMA packet-radio network architecture.

A noteworthy example is the end-to-end flow-control algorithm. Our CDMA network architecture uses a new packet-radio congestion-control algorithm[6] (Appendix A) that dynamically adjusts end-to-end flow rates. The throughput and survivability improvements we report are due mostly to the link-layer algorithms, whereas the fairness improvements are due mostly to the end-to-end congestion-control. We will indicate the source of improvement whenever possible. In addition we are not the architects of the link-gain-adjustment algorithms (Appendix B) that determine power levels, FEC rates and bit rates. We mention these algorithms briefly for the sake of completeness and refer the reader to a more thorough paper [7] on this topic.

Our paper is organized in the following way. In Chapter (2) we explain the fundamental performance factors and the basic link-layer design issues for packet radio networks. Then we give an overview of our CDMA network architecture followed by a summary of SURAP. We conclude this chapter with a comparison of the two architectures that introduces the performance issues we investigated with our simulation. Chapter (3) gives the results of our performance study and Chapter (4) summarizes these results and makes recommendations for future research.

2. Link-Layer Protocols for Packet-Radio Networks

A link-layer protocol provides reliable internode data-packet transport by:

1. retransmitting errored or blocked data packet transmissions,
2. adjusting link gains to control bit-error rates,
3. adjusting link flow rates to:
   (a) stop the spread of congestion, and
   (b) maintain channel stability.

These vital functions are achieved by distributed algorithms that use control information obtained from a dialogue that consists of transmitting and receiving data and acknowledgement packets. Designing a link-layer protocol for packet-radio networks is a formidable challenge because the links are often unreliable, their bit-error rates are highly nonstationary, and packet transmissions are frequently blocked.

In the next section we explain the performance factors that impact link-layer design for packet radio networks. Later, we outline our CDMA and the SURAP link-layer protocols and conclude this section with a comparison of the two designs.

2.1 Packet-Radio-Network Performance Factors

An important concept that differentiates packet-radio networks from other multiple-access networks, e.g. Ethernet LANs, is *spatial reuse*. Each packet radio belongs to a *neighborhood* that contains all the packet radios in its radio range. When a packet-radio network spans a large geographic region and there are packet radios that are out of range of each other because of signal attenuation, the region will be covered by overlapping neighborhoods. Then the common radio channel has spatial reuse in the sense that a packet radio can communicate with any radios in its own neighborhood without affecting packet radios in other isolated neighborhoods.

If a packet radio network did not employ spatial reuse all the radios would be in the same neighborhood and the network would be topologically equivalent to a LAN. In most cases maintaining this topology would require transmissions at very high power levels. But a packet radio cannot receive packets when overlapping transmissions have too much power. This can lower the signal-to-noise level at a packet radio so much that it loses synchronization or the received packet is so corrupted with bit errors that FEC algorithms are ineffective. This performance impairment is called *mutual interference*.

The uncoupling created by spatial reuse can increase a network's overall throughput by increasing the number of simultaneous transmissions and decreasing mutual interference. But, as we will soon show,

spatial reuse creates undesirable interactions between overlapping neighborhoods that decrease network throughput.

Consider Figure 2.1 where an arrow between two nodes indicates they are in mutual radio range. The indicated connectivity implies packet radio A can hear the transmissions of radios X, Y, Z, and B but neither C nor D. The first four are in A's neighborhood; the latter two are not. We say that radios C and D are *hidden* from radio A and radio A is *hidden* from radios C and D. In general radios that are not in range are hidden from each other. Thus whenever packet-radio node D is transmitting (to node C), node

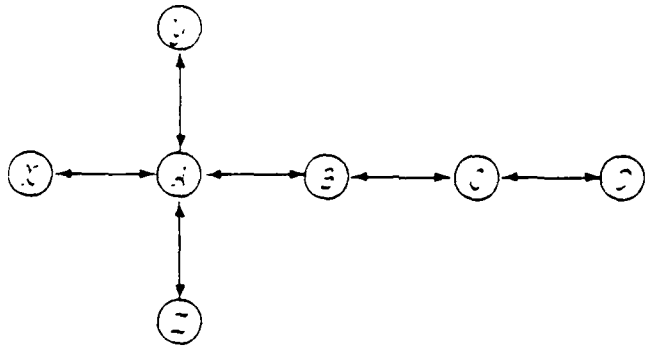


Figure 2.1: Packet-Radio-Network Connectivity

B will not receive these transmissions and instead it can receive transmissions from node A— concurrent successful transmissions increase network throughput. On the other hand, a packet radio cannot receive two transmissions at once neither can it receive while it is transmitting, but such concurrent transmissions can occur whenever there are hidden packet radio nodes. For example, whenever packet radio C is receiving from packet radio B, packet radio D cannot detect this and it may transmit (unsuccessfully) to packet radio C. We say these transmissions are blocked. Blocking wastes bandwidth and the ineffective radio transmissions increase mutual interference.

CDMA and broadcast networks are both affected by blocking and mutual interference to varying degrees but the precise way these performance factors impact network performance is difficult to quantify.

In a previous paper Zavgren and Lauer [8] simulated packet-radio node placement and routing to determine the mapping between end-to-end flows and link-layer flows for randomly-generated network topologies. Then they applied an analytic model to particular topologies to determine these networks' maximum aggregate end-to-end throughputs (capacity) and the mutual interference generated noise-power level at individual nodes. Their work showed that a CDMA network can have five times the capacity of an identical network with a broadcast channel; furthermore, CDMA networks are less susceptible to jamming than broadcast networks because their lower blocking levels require fewer retransmissions, decreasing mutual interference. The performance gain is topology dependent and closed-form analytic results are only obtainable when networks with significant (i.e., atypical and unrealistic) symmetry are analyzed. Nevertheless, we can conclude that the denser the packet radio placement, the greater the relative throughput improvement. The simulation results in Section (3) are consistent with this earlier study, however the magnitude of the performance gains are less than predicted in some cases because the earlier study did not include the effects of end-to-end flow control. Aggregate end-to-end throughput

can be decreased when max/min fair end-to-end flow control is provided.

The following two sections outline the CDMA and the SURAP link layer protocols and explain two important components for each: link-level acknowledgements, and link-level congestion avoidance. Link-level acknowledgements are used to verify the reception of individual packets; unacknowledged transmissions are repeated. The design of the mechanisms for transmitting and receiving link-level acknowledgements is very tightly coupled to the physical layer. CDMA networks require the transmission of a separate acknowledgement packet — called an active acknowledgement — whereas broadcast networks use a more subtle mechanism — called a passive acknowledgement that's based on overheard data packet transmissions. Link-level congestion-avoidance algorithms are designed to prevent the spread of congestion by temporarily restricting the utilization of critical resources, e.g., packet buffers and the radio channel.

2.2 CDMA Networks

2.2.1 Link-Level Acknowledgements

Our CDMA link-level protocol is based on the assumption that every packet radio is assigned a unique spreading code that it uses for receiving data and acknowledgement packets — transmissions on other spreading codes cannot be detected. This section gives the protocol that controls the exchange of data and acknowledgement packets between packet radios. Its based on the Preemptable Moment of Silence Algorithm [9], which uses two basic principles:

- It forces a data packet transmitter to defer all additional data-packet transmissions until it has monitored the channel for a short time interval called a Moment of Silence.
- It forces a data packet receiver to expedite its formulation of an acknowledgement packet and to transmit this packet at a priority level that exceeds that of all data packets.

As a result, the transmitter of a data packet gives higher priority to receiving an acknowledgement than to sending another data packet and the receiver of a data packet is more likely to transmit its acknowledgement packet when the data-packet transmitter is listening. This algorithm is called the Preemptable Moment of Silence algorithm because either node can have its Moment of Silence preempted by a data packet reception.

Figure 2.2 illustrates a successful data-packet transmission. The parameter τ_{max} establishes the maximum duration of a Moment of Silence. If an acknowledgement is not received before τ_{max} time units have elapsed, the Moment of Silence is terminated. Otherwise, it ends with the reception of an acknowledgement packet. The value of τ_{max} is a design parameter that depends on the distribution of processing time. Large values of τ_{max} decrease both the acknowledgement-packet blocking rate and the maximum data-packet transmission rate — small values increase both. Because the data-packet processing times in the LPR are essentially deterministic, the optimum setting is slightly larger than the sum of the minimum data-packet processing time and the minimum acknowledgement-packet transmission time.

The Moment of Silence does not prevent acknowledgement-packet blocking but it does make the probability of this event extremely small. Because acknowledgement packets and data packets are received on the same spreading codes, occasionally a node's Moment of Silence will be preempted by a data-packet

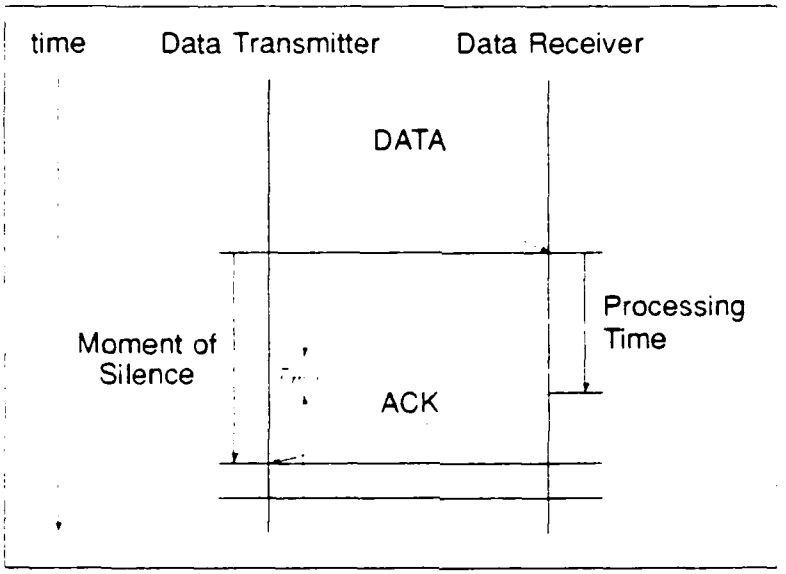


Figure 2.2: An Acknowledged Data Packet Transmission

reception that blocks an acknowledgement reception. But this happens infrequently and when it does the data packet transmission is repeated.

The Moment of Silence algorithm produces an efficient link-layer. It increases throughput by making the blocking rate for acknowledgement transmissions extremely low. A useful by-product of this algorithm is a low-delay high-reliability "conduit" for acknowledgements. We have utilized this property by generalizing the concept of an acknowledgement packet into a *control packet*. These packets not only acknowledge the reception of data packets to provide a reliable link, they also are used for flow control, link-gain adjustment, and the distribution of end-to-end congestion control information[6]. We define three types of control packets: acknowledgment packets, flow-control packets, and gain-adjustment packets. Exactly one of these packets is transmitted after every data-packet reception. When a packet radio receives a data packet and:

- buffers it (i.e., all bit errors are correctable and a buffer is available), it transmits an *acknowledgment packet*
- discards it because it has no buffers available, it transmits a *flow control packet*
- discards it because it cannot be error-corrected, it transmits a *gain-adjustment packet*. This is the case if a packet header is received without errors but the body is not. (We advocate the encapsulation of the ID of the sender with a strong error-correcting code so gain adjustment packets can be transmitted when needed, but this feature is not included in our simulation.)

Thus, the type of control packet transmitted is determined by what a packet radio does with each received data packet. A control-packet summary is provided in Table (2.1).

Link-Level Control Packets	
Type	Purpose
Acknowledgement	Acknowledges the reception and buffering of the neighbor's data packet.
Flow Control	Informs the neighbor that its data packet was received but discarded due to insufficient buffers.
Gain Adjustment	Informs the neighbor its data packet was received but could not be error corrected and was discarded.

Table 2.1: Link-Level Control Packets

2.2.2 Congestion Avoidance (New CDMA Algorithms)

We employ two cooperating congestion-avoidance mechanisms. One is an *end-to-end* flow-control algorithm that establishes long-term — relative to the length of a few packet transmissions — max/min fair end-to-end flow rates, that is summarized in Appendix A. In this section we explain the *link-level* congestion-avoidance algorithms that serve three important purposes: facilitate the flow of data packets, protect critical nodal resources from overload during short-lived traffic bursts, and provide a "safety net" to prevent the spread of congestion when the end-to-end flow-control algorithm is adapting to a change in traffic patterns.

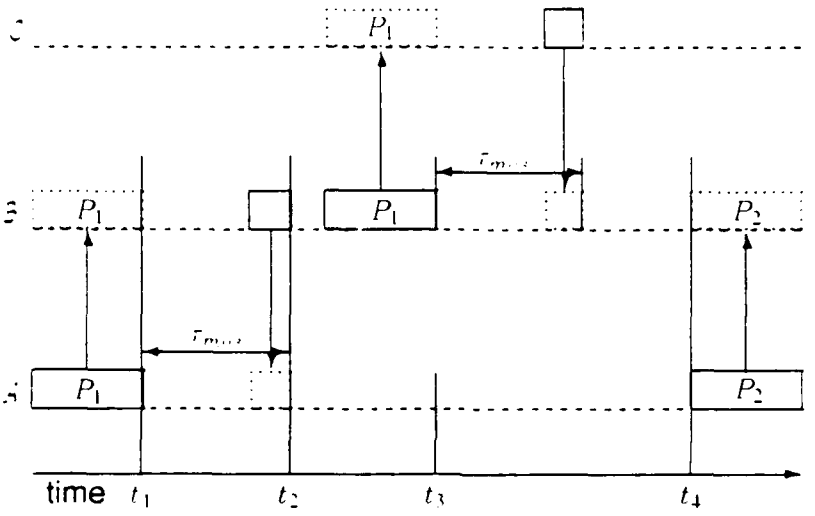
The relationship between these two congestion-avoidance mechanisms has two aspects. First, the end-to-end mechanism takes long-term actions that anticipate the onset of congestion to prevent it from occuring, but the link-level mechanism takes short-term actions that react to congestion and quench it. Second, the end-to-end mechanism regulates the flow of packets, whereas the link-level mechanism in its normal mode of operation, expedites the flow of packets.

The CDMA network uses its link-level congestion-avoidance mechanisms *only* when network resources are exhausted. Otherwise — that is, if link-level mechanisms are used to establish link-level flow rates — the requirements of link-layer flows would often conflict with those of end-to-end flows, resulting in unfair end-to-end flow allocations. Also, this would bias the measurements used by the end-to-end algorithm and result in even more unfairness. The CDMA link-level congestion-avoidance mechanisms do not alleviate congestion; instead, they temporarily restrict access to resources so that congestion does not spread to other portions of the network.

Because the bandwidth-delay product for the LPRnet is small, our algorithms use a link-level window size of one, that is, a node is inhibited from transmitting a data packet to a neighbor until it receives an acknowledgement for its most-recent data-packet transmission or this packet is discarded. Our three link-level mechanisms are spacing, buffer management and retransmissions.

Spacing The spacing algorithm sets an upper bound on the rate every node transmits data packets to each of its neighbors. A spacing computation follows each control-packet reception to determine the earliest time the next data packet can be transmitted to this neighbor. The objective is to provide sufficient pause between transmissions to protect a neighbor node's processor from overload and to avoid collisions with the neighbor's transmission of the packet it just received.

A spacing calculation is illustrated in Figure 2.3, where node A transmits two packets, P_1 and P_2 to node B, which, node B then transmits to node C. Dotted boxes represent packet receptions, solid boxes packet transmissions, long boxes data packets, and short boxes control packets.



t_1 Node B receives packet P_1 from node B.

t_2 Node A receives node B's control packet (an acknowledgement).

t_3 Node A has processed this acknowledgement to the extent that it must compute the earliest time of its next data transmission to node B i.e., time t_5.

t_4 Node A transmits P_2 to node B.

Figure 2.3: A Spacing Computation

When node B receives packet P_1 it informs node A (in its acknowledgement packet) of the transmission time of packet P_1 to node C. Node A requires this information because P_1's transmission time by node B depends on node B's FEC and bit rates. Then node A determines the earliest time it can send packet P_2 to node B by estimating the time that node B completes the processing of its acknowledgement from node C. This estimation is done with the following formula:

$$t_4 = t_3 + \tau_{m,s} + P_{ack}, \tag{2.1}$$

where P_{ack} is the mean control packet processing time. This computation assumes that time t_3 and

node B's transmission of packet P_1 coincide, a reasonable assumption when packets are not queued for processing at either node. The algorithm in Equation (1) works well for the LPR.

We've investigated more sophisticated spacing algorithms but found they were not worth implementing. For example, we analyzed an algorithm that required node B to tell node A how long it would be transmitting packets (including moments of silence) in its next sequence of contiguous transmissions. But this algorithm is difficult to implement and the estimate is flawed by receptions and unacknowledged transmissions, resulting in only a marginal performance improvement.

Buffer Management The K-buffering algorithm limits the number of packets that a node will buffer for any neighbor to not exceed the value set by the parameter K. When a node receives a data packet from a neighbor for which K packets are buffered or when all its packet buffers have been allocated, this packet is discarded and a *flow-control packet* is transmitted to inform this neighbor how long to wait before it may transmit the packet again. The first time a packet is discarded the neighbor is given a waiting time equal to the elapsed time between the first transmission attempt and buffer deallocation, smoothed over all data packets that have been forwarded for this neighbor. Thereafter, the flow control packets contain waiting times that increase exponentially (until an upper limit is reached) until a buffer is deallocated or the packet is discarded. This flow-control action reduces packet discarding by forcing particular neighbors to reduce their transmission rates until more buffers are available for them, thereby reducing mutual interference and processing loads.

The value chosen for K has a large impact on throughput and fairness. Figure 2.4 shows a network with two, two-hop routes (indicated by solid arrows) that share the same first-hop link but use different second-hop links with different bandwidths. In general, the maximum end-to-end flow rate for any route

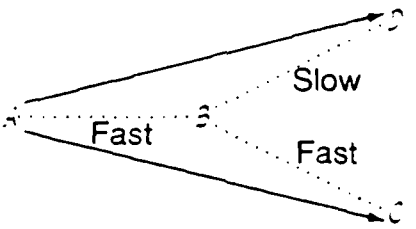


Figure 2.4: The Passing-Lane Effect

is bounded above by the speed of its slowest link, but interactions between routes can significantly decrease the actual flow rates.

Consider the case when $K = 1$. Then when node B buffers a data packet destined for node D it will not buffer any more packets from node A until it receives an acknowledgement from node D (or it discards the packet after six tries). Thus traffic on the $A \Rightarrow D$ route can restrict traffic on the $A \Rightarrow C$ route. Now consider the case when K is larger. Then when node B buffers a data packet destined for node D it can buffer another packet from node A if the total number of packets buffered for node A is less than K. If the flow rate on each route is set to the speed of its slowest link by the end-to-end flow control algorithm, then a moderately large value of K can virtually eliminate the interactions between traffic on the two routes, and increase the flow on the $A \Rightarrow C$ route to its maximum rate.

The case of $K = 1$ is analogous to a one-lane highway, whereas the case with larger K is analogous to a highway with passing lanes. A small value of K not only lowers the throughput of some routes, it also biases the measurements given to the end-to-end flow-control algorithm and results in unfair end-to-end flow allocations. To avoid these adverse interactions between the link-level congestion-avoidance algorithms and the end-to-end flow control algorithm we advocate the use of a large K value, even though this can result in a node ocassionally allocating a disproportionate number of buffers to a few neighbors for short time periods. The LPR has eleven packet buffers; we have set K to seven and to prevent deadlocks, we attempt to keep two buffers in reserve at all times for receiving packets.

Retransmissions The retransmission algorithm is used by a node to set the transmission times to a neighbor that is not returning control packets, either because it is blocking data packet transmissions or because it cannot decode the part of the packets that contains the sender ID and return a gain adjustment packet. The (conflicting) objectives of the algorithm are fourfold:

1. increase throughput by retransmitting quickly
2. maintain channel stability by retransmitting slowly
3. decrease the blocking rate by staggering transmissions when more than one node is retransmitting to the same node
4. prevent retransmissions from occuring prematurely, i.e., before control packets can be transmitted.

If the first item were not a concern, the design of the retransmission algorithm would be quite simple. In this case we would use a random backoff algorithm with a large average delay, e.g., an algorithm that chooses retransmission times so their values are uniformly distributed on time intervals that increase rapidly in length with the number of consecutive unacknowledged retransmissions. But because blocking occurs frequently, this conservative approach introduces too much delay and consequently reduces throughput. Instead, we have taken the following more aggressive approach.

The retransmission algorithm inserts sufficient pause between retransmissions to make the event that a retransmission occurs before a control packet can be transmitted extremely rare. It achieves this goal by using a constant bias equal to T_{xmit}. The first retransmission time increment is given by:

$$T_1 = T_{xmit} + D_1$$

where $\{D_i\}$ is a sequence of independent random variables that are uniformly distributed over a time interval given by the sum of a Moment of Silence and the maximum packet transmission time. With each subsequent retry the intertransmission time increment increases *linearly* in mean:

$$T_{n+1} = T_{xmit} + n \cdot D_{n+1} \text{ for } n = 1, 2, \ldots, 5$$

We call the resulting algorithm linear backoff.

When a gain adjustment packet is received a retransmission is also needed, but the requirements for scheduling this retransmission are quite different because the cause is channel noise, not blocking. In these cases the link gain is adjusted appropriately and the packet is retransmitted after a very short random delay.

2.2.3 Congestion Avoidance (Algorithms from SURAP)

The remaining three algorithms are SURAP algorithms that we have used in our CDMA network architecture.

Fairness Queueing The LPR has a single Transmit Queue that contains packets waiting to be transmitted. Packets are processed FIFO as they are received and inserted in this queue; the transmissions however are not necessarily FIFO. There are two exceptions.

First, a packet is never transmitted until its *transmission timer* expires. This timer is set when a packet is first processed and reset with each transmission. Because the pacing (used only in SURAP, see Section 2.3.2) and spacing (used only in CDMA, see Section 2.2.2) algorithms can set different transmission rates to different neighbors a packet going out a "fast" link can be transmitted before a packet going out a "slow" link, even when the packet going out the slow link arrived first. This feature enhances the passing lane effect illustrated in Figure 2.4.

Second, link reliability can vary greatly in a packet radio network; with straight FIFO queueing, the more reliable links could monopolize buffer and channel resources, creating unfairness. When packets received from more than one neighbor are queued with the same "next-hop" destination, *fairness queueing* will not allow consecutive packet transmissions from the same neighbor to this destination. In these cases the order of the packet transmissions is modified so packets will be transmitted, round robin, on a neighbor-by-neighbor basis. This prevents a neighbor with a reliable link, or a host with a wire interface from getting an unfair bandwidth allocation.

Alternate Routing After a packet has been transmitted four times without an acknowledgement a special "alternate-routing" flag is set in the packet header and it is broadcast on the last two transmissions to request help in forwarding the packet. Normally packet radios will never transmit received data packets that are not addressed to them. But when the alternate-routing flag is set and the destination is listed in their routing tables with a shorter distance they attempt to forward the packet. This feature allows packets to be quickly rerouted around temporary link failures. If the problem persists, the packet radios will update their routing tables to eliminate the link.

When nodes in a CDMA network hear an alternate-routed packet, they can bombard the transmitter with overlapping acknowledgement transmissions and the mutual interference could prevent a correct reception of any of them. We've modified the SURAP alternate-routing algorithm for CDMA to avoid this flood by randomizing the transmission time of acknowledgements to alternate-routed packets. The randomization interval is of the order of a Moment of Silence.

Packet Dropping Both protocols set a limit of six on the number of times a node can transmit a data packet without receiving an acknowledgement. If on the sixth transmission attempt no acknowledgement is received, the packet is discarded and higher-level algorithms may resubmit it to the network at a later time. Packet dropping is an important mechanism for purging a network of undeliverable traffic to prevent deadlock when it is congested.

2.2.4 Routing

Routes are calculated in the LPRnet with a distributed minimum-hop algorithm, called tier routing. The tier level of a destination is the number of hops required to reach it. This algorithm requires each node to maintain a tier table that lists which packet radio is the next hop enroute to each reachable destination. Each LPR broadcasts its tier table at an average rate of once per 7.5 seconds in a special packet called a PROP. As packet radios receive PROPs they learn of shorter routes to old destinations and routes to new destinations and update their tier tables accordingly.

2.3 Broadcast Networks

We describe the link-layer protocol used in SURAP. Variations of this design are used by SURAN, SINCGARS [5] and an experimental system for library automation [10]. A detailed description of SURAP is published in a paper by Jubin and Tornow [2], however the reader must be aware that the current SURAP algorithms deviate slightly from this reference. The only noteworthy impact on our performance comparison was the elimination of "quadratic backoff" which has been demonstrated to cause unfair end-to-end flow allocations[11]. This and other less significant modifications to SURAP have been incorporated into our broadcast-network simulation. In spite of these changes [2] is still an excellent introduction to broadcast link-layer design and other related packet-radio algorithms, e.g., routing.

2.3.1 Link-Level Acknowledgements

A fundamental concept in broadcast link-layer design is the *passive acknowledgement*. A broadcast channel allows any node in range of a packet transmission to receive it. For example, when node B (Figure 2.5) transmits in an attempt to forward a data packet to node C, both nodes A and C can receive this transmission. Node C will process the packet and prepare it for transmission to node D, whereas node A will recognizes the packet as an earlier transmission to node B. To node C it is a data packet reception; to node A it is a passive acknowledgement. A single transmission serves both purposes.

Note that there are two cases when passive acknowledgements cannot be used. A packet radio (e.g. node D) will not use its radio transmitter to forward packets it receives as the final destination of a route. Instead it will send them over a wire interface to an attached host. In these cases a separate, or *active* acknowledgement transmission is required. This is illustrated in Figure (2.5) by node D transmitting an active acknowledgement packet. The other situation that requires an active acknowledgement arises when a passive acknowledgement is blocked or corrupted with bit errors. If, for example, node A in Figure (2.5) does not receive node B's passive acknowledgement, node A will repeat the data transmission and node B will respond with an active acknowledgement if it receives the packet.

2.3.2 Congestion Avoidance

Single Threading The single-threading policy outlined by Jubin and Tornow is equivalent to a link-level window size of one, meaning a policy of not transmitting a new packet to a neighbor until the most recent transmission has been acknowledged. When single threading is combined with passive

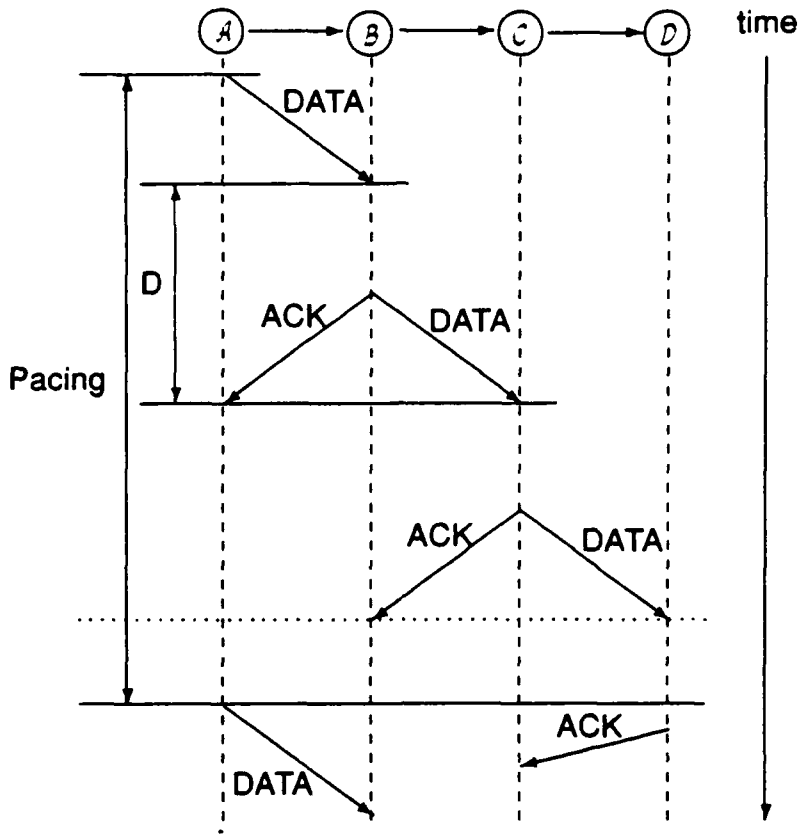


Figure 2.5: An Illustration of Passive Acknowledgements and Pacing

acknowledgements, the resulting policy implicitly limits the number of packets from each neighbor that a packet radio will buffer. For example, suppose that node A in Figure (2.5) transmits a data packet to node B and the packet's next destination after B is C. Here B will not acknowledge this packet — or equivalently, will not transmit the packet to C — until any previous packets it has transmitted to C are acknowledged. Hence, the combined policy tends to have radios buffer only one packet for each radio forwarding through it. The objective of single threading is to limit the number of packets in the subnet and to create "backpressure" which prevents packets from entering the network until it has adequate free resources.

Pacing The Pacing algorithm sets packet inter-transmission times on a neighbor-by-neighbor basis. Whenever a packet radio transmits a particular data packet it records the time the transmission ends. When an acknowledgement is received for this transmission, the time the acknowledgement reception concludes is also recorded. The difference between the two times (labeled D in Figure (2.5)) is called the *forwarding delay*. It includes processing, queueing and transmission delays at the receiving node. Forwarding times are measured and smoothed by each node on a neighbor-by-neighbor basis to maintain

a short-term history of the delay through each neighbor.

The *pacing delay* (labeled Pacing in Figure (2.5)), on the other hand, determines the time between successive data transmissions to each neighbor. Its defined as three times the forwarding delay since the receiving node must receive, transmit and receive an acknowledgement for each packet[3]. The dashed line indicates the earliest time node A can transmit; because of forwarding delay variance the pacing delay will occassionally overestimate or underestimate the minimum inter-transmission time.

2.4 Architecture Comparison

In this Section we summarize the differences between the SURAP and the CDMA protocol suites and introduce the performance issues addressed in Section 3.

2.4.1 Blocking and Mutual Interference

As shown in [8], CDMA networks can generally sustain a higher throughput because they allow concurrent packet transmissions to different destinations in the same radio neighborhood. However, for some network configurations the mutual interference caused by these additional transmissions can increase the bit error rate and decrease throughput. Broadcast networks, on the other hand, tend to prohibit concurrent transmissions because whenever a packet radio transmits, every idle packet radio in radio range locks onto the transmission, thus preventing all from transmitting until the transmission is over. This staggering of transmissions reduces mutual interference. The mutual interference difference is most profound when the network has no hidden terminals, and connectivity is high, e.g. Figure 3.6.

The reduced mutual interference in broadcast networks tends to increase throughput because the lower bit error rates decrease the retransmission rate. However the transmission staggering in broadcast networks decreases the rate the individual radios can transmit data packets and thus decreases throughput. Also packet radios in a broadcast network can block transmissions when they receive "unwanted packets"; this effect is quite noticable when there are hidden terminals, e.g., Figure 3.3.

2.4.2 Acknowledgement Traffic and Power Control

In the absence of blocking and mutual interference, an N-hop route in a broadcast network requires $N + 1$ transmissions: N data packet transmissions and one active acknowledgement transmission. The same route in a CDMA network requires $2 \times N$ transmissions: N data transmissions and N active acknowledgement transmissions. For example, in Figure 2.5 three data packet transmissions and one active acknowledgement transmission are needed to forward one data packet along a three hop route. The same route in a CDMA network would require three data packet and three active acknowledgement packet transmissions. Based on transmission counts alone, it seems broadcast networks have a more efficient channel that can allocate more bandwidth to data packet transmissions because less bandwidth is required for acknowledgements.

This efficiency argument is valid when there is no blocking and power control is not considered but it is not true in general. When a broadcast channel sends a data packet and a passive acknowledgement with the same transmission, the power levels needed to reach the two nodes can differ by several orders of magnitude. In general, whenever passive acknowledgements are used on a broadcast channel each

packet must be transmitted at the maximum of the two power levels. When the acknowledgement power requirement is higher, a shorter active acknowledgement transmission at this power level uses less energy and thus reduces mutual interference. For some configurations higher throughputs could be achieved with a broadcast channel if active acknowledgements were used.

2.4.3 Transmission Scheduling and Flow Control

In a broadcast network the passive acknowledgement transmissions must be scheduled with the same algorithm as data packet transmissions because one transmission serves two purposes. The scheduling objectives for data transmissions and acknowledgment transmissions are in conflict. Data packet transmissions must be scheduled to achieve flow control, channel stability, etc., whereas acknowledgement transmissions should be scheduled to free buffers quickly. Bundling the two transmissions together with the use of passive acknowledgements necessitates a performance compromise. The SURAP Pacing algorithm is designed for congestion avoidance, not for quick reliable acknowledgement transmissions.

The CDMA network, on the other hand, transmits separate acknowledgement and data packets. The Moment of Silence algorithm keeps the acknowledgement blocking levels very low and the Spacing algorithm schedules data packet transmissions to reduce blocking while expediting the flow of data packets. This twofold strategy significantly improves throughput by decreasing data and acknowledgement packet blocking.

The Pacing algorithm used by SURAP determines the time between a node's transmissions to each of its neighbors by measuring their forwarding delays. But because of changes in link reliability and packet transmission times these statistics can have significant variance, resulting in the Pacing algorithm over- and under-estimating pacing delay. This increases blocking and decreases throughput.

The Spacing algorithm, on the other hand does not require a node to measure the duration of its neighbors transmissions, instead each successful transmission is answered with a control packet that specifies the duration of the next transmission of the packet on the next leg of its journey. This up-to-date information allows a node to schedule its transmissions more accurately, thereby reducing blocking.

2.4.4 Fairness

SURAP sets end-to-end flow rates with a combination of Pacing and Single-Threading. This strategy allows each packet radio to set the flow rates on each of its links unilaterally which can result in unfair end-to-end flow rates.

The CDMA protocol suite, on the other hand, employs two cooperating congestion-avoidance mechanisms, an *end-to-end* flow-control algorithm that establishes long-term — relative to the length of a few packet transmissions — max/min fair end-to-end flow rates, and *link-level* congestion-avoidance algorithms, e.g., K-buffering and Spacing that facilitate the flow of data packets and provide "passing lanes", protect critical nodal resources from overload during traffic bursts, and provide a "safety net" to prevent the spread of congestion when the end-to-end flow-control algorithm is adapting to a change in traffic patterns.

3. Performance Analysis

We present simulation results that compare the performance of SURAP and the CDMA protocol suites. Because packet-radio network performance is a complex function of topology, node proximity, and network traffic patterns it is infeasible to investigate all these effects in networks of significant size by simulation. Therefore, we have taken the approach of identifying representative "topological components" of packet-radio networks, e.g., splits, merges, and parallel flows with radio interference, that test a network's throughput and fairness while maintaining a feasible number of nodes. We ran extensive simulations with these components to improve our design and tune its parameters. After we completed this exploratory work, we repeated the simulations with the SURAP algorithms to establish a comparison between the two network architectures. We report some surprising and dramatic performance differences.

3.1 Background Information

Our results are from a discrete-event simulation that models radio interference. As a result were able to invesigate not only the network performance impact from link-level algorithms but the affects of radio interference as well.

The radio interference model computes the signal-to-noise ratio at each receiver for each of its packet receptions, by including the affects of antenna gain, processing gain, and path loss. The resulting signal-to-noise ratio is mapped into a bit-error rate by the use of the Gaussian Q function then this rate is mapped into the number of bit errors in each packet by an efficient Monte-Carlo technique. After the number of bit errors have been computed, a heuristic approach is used to compute the probability that the errors can be corrected by the convolutional FEC decoder. This model is discussed in detail in reference[7].

The discrete event simulation assumes heavy traffic, i.e., all route sources send traffic at the highest rate allowed by the network protocols. Packet lengths are independent and uniformly distributed between 528 and 2428 bits, including headers, for a mean packet length of approximately 1500 bits.

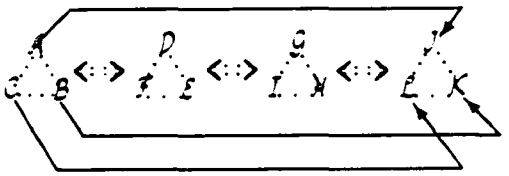
The LPR has two processors, one that operates the transceiver and one for processing packets. We model processing times in the following way. There is a one millisecond delay in the LPR when the transceiver gives a packet to the packet processor. The packet processor takes a minimum of five milliseconds to process a packet to be sent over the wire interface and a minimum of eight milliseconds if the packet is to be transmitted on the radio channel. Either of these two processing tasks can take longer when interrupts occur or transmission timers expire. Each of these events takes two milliseconds and on average one occurs per each received packet. Thus, the minimum processing time for a received packet that is sent over the wire interface is six milliseconds and the minimum processing time otherwise is nine milliseconds. Average processing times for the two cases are roughly eight and eleven milliseconds, respectively. Acknowledgements take the same amount of time to process as data packets that are sent over the wire interface.

We have organized our simulation results into four catagories: Blocking and Mutual Interference, Jamming, Acknowledgement Effects, Fairness, and Fault Recovery. The first three categories contain experiments designed to investigate physical and link-level issues. The latter two categories are concerned with higher-level or systematic issues. These results are presented in the following ten subsections. Each one has a separate panel that contains a figures depicting the network topology and plot of simulation results. The convention for showing network topology is that dotted lines show radio connectivity and solid, arrowed lines show routes. Each panel is followed by a brief experimental description.

Each simulation result shows throughput as a function of time. We have rerun each simulation with several random-number-generator "seed values" and we have determined that the seed value used for generating a throughput curve does not have a significant impact on our results.

3.2 Blocking, and Mutual Interference

3.2.1 Parallel Flows



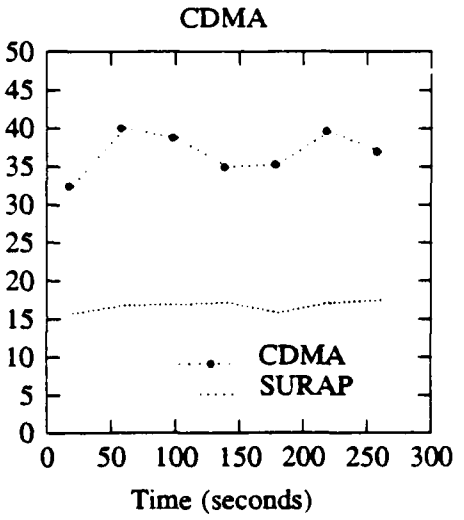


Figure 3.1: Aggregate Throughput

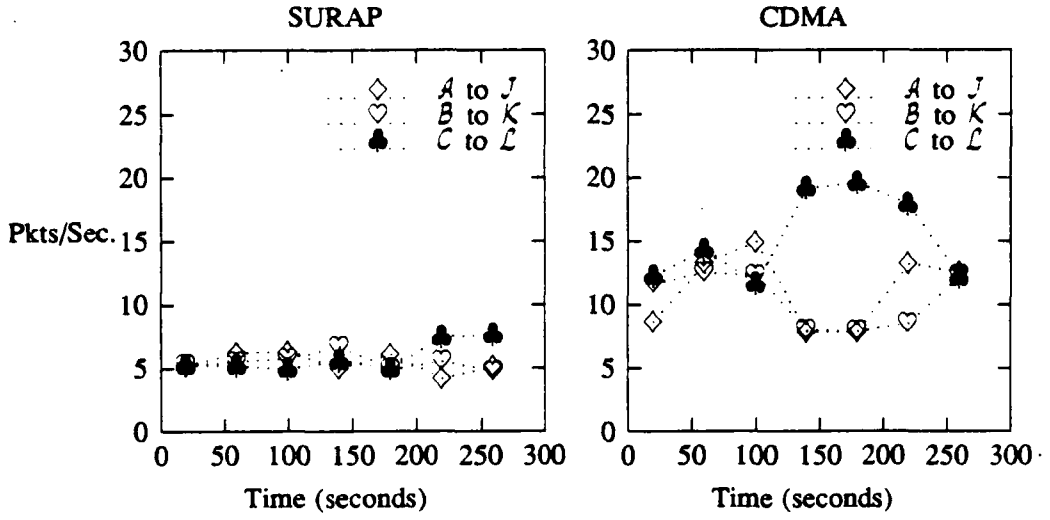


Figure 3.2: Route-by-Route Throughput

This experiment was designed to show the effect of Code Division on network throughput. In order to simplify the network diagram we lumped nodes into clusters. For example nodes A, B, and C are in the same cluster. Every node is in radio connectivity with each node in its cluster and each node in every adjacent cluster.

Figure 3.1 shows the aggregate throughput for the two protocols suites. Note that the CDMA algorithms deliver more than twice the throughput of SURAP. If the three routes had disjoint paths the throughput improvement would be approximately three-fold but because of interactions between routes the throughput is lower. Notice that between 125 and 225 seconds into the simulation the $C \Longrightarrow L$ route approximately doubles its throughput, while the other two roughly halve their throughput. On closer examination, we determined that during this time interval the $A \Longrightarrow J$ and the $B \Longrightarrow K$ route intersect at node D, while the other route follows a disjoint path. Nevertheless, this bandwidth allocation is fair, given the particular (suboptimal) routing.

3.2.2 Hidden Terminals (Part I)

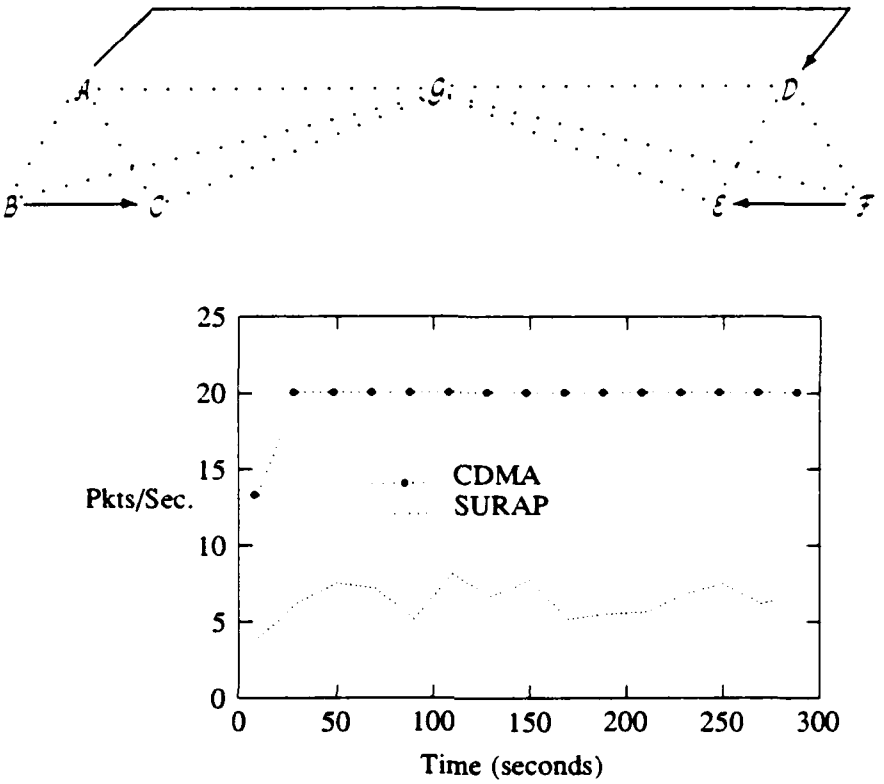


Figure 3.3: $A \Rightarrow D$ Throughput

This simulation demonstrates how hidden terminals can reduce network throughput. Nodes A, B, and C are hidden from nodes D, E, and F. Node G on the other hand can hear every node in the network. Our simulation showw that the CDMA version of this network has about four times the throughput of the SURAP version. The throughput reduction with SURAP is because of data and acknwowledgement packet blocking. We anticipate this network configuration to occur frequently, especially when repeaters are placed on the tops of hills to connect hidden radios.

3.2.3 Hidden Terminals (Part II)

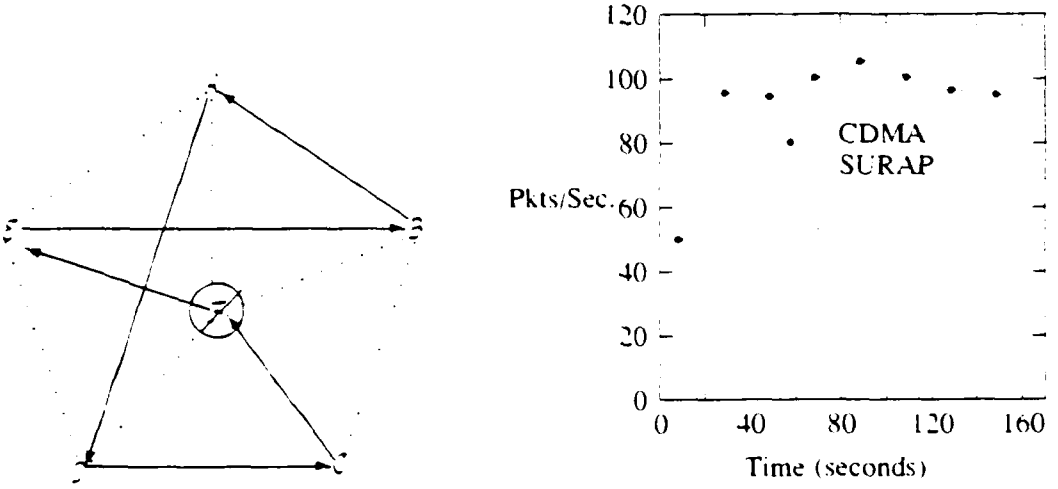


Figure 3.4: Aggregate Throughput

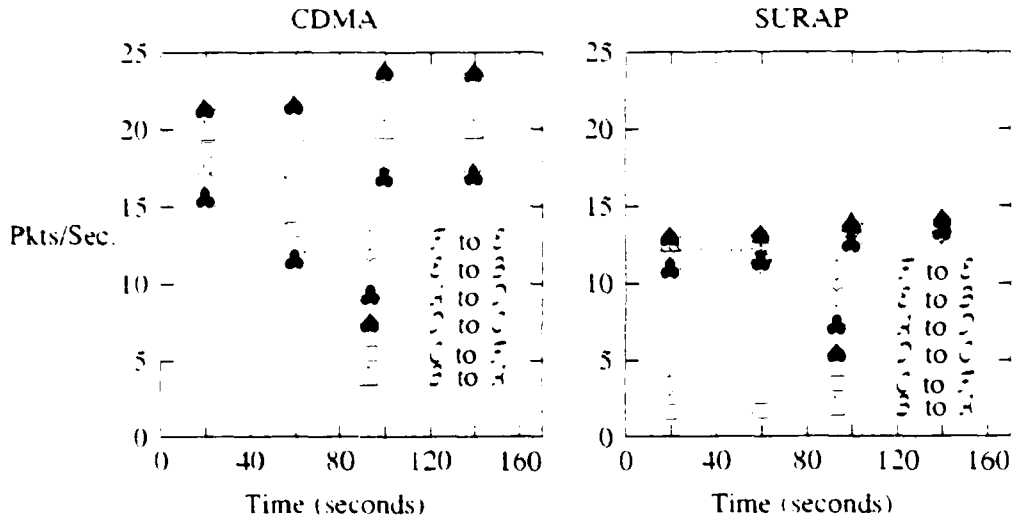


Figure 3.5: Route-by-Route Throughput

This simulation illustrates problems that occur when a single node listens to many hidden terminals. The central node $\mathcal{F}$ can overhear the traffic from all radios many of which cannot hear each other; with SURAP this causes it to block packets at a very high rate. This is why the $\mathcal{F} \Longrightarrow \mathcal{E}$ and $\mathcal{C} \Longrightarrow \mathcal{F}$ routes achieve such low throughputs with SURAP.

The central node $\mathcal{F}$ stops operating one hundred seconds into the simulation. This reduces the CDMA network throughput but has no significant affect on the SURAP throughput, indicating that the CDMA network is more effeciently utilizing its available resources. Or, adding resources to a SURAP network

may not increase its capacity. The aggregate throughput for the CDMA network is about twice that of the SURAP network.

3.3 Jamming

3.3.1 High Connectivity

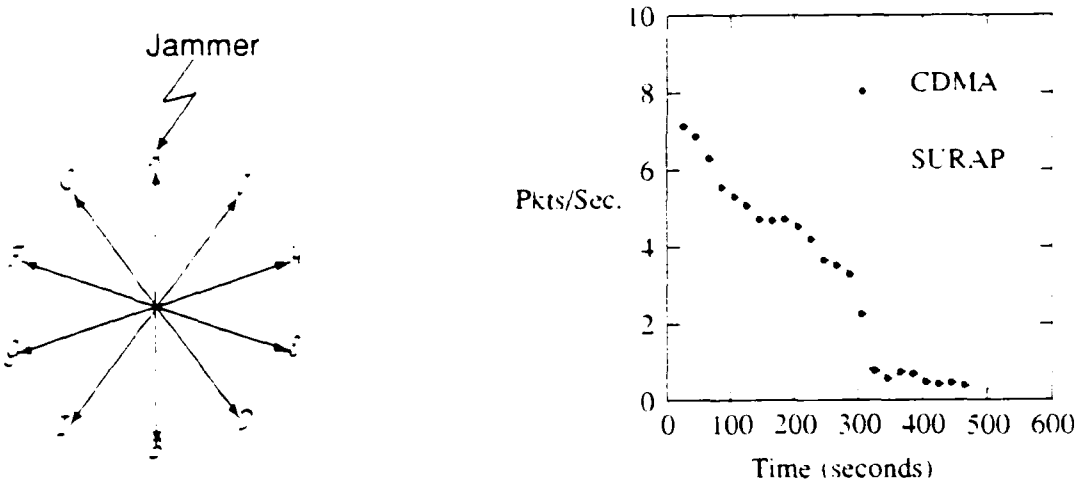


Figure 3.6: Jamming Effect on $\hat{5} \Leftrightarrow \hat{2}$ Throughput

We simulated ten packet radio nodes in mutual radio connectivity. Node $\hat{5}$ is attacked by a jammer. The jamming power increases linearly over the duration of the experiment, starting at .5 picowatts and ending at .62 picowatts.

When the jamming power is low the CDMA network has about three times the throughput of the SURAP network, because the former allows simultaneous packet transmissions to different destinations. For this particular example the SURAP network has no packet blocking because every packet transmission is received by every packet radio, preventing them from transmitting at the same time. This staggering reduces mutual interference.

As the jamming power increases, the throughput of the CDMA network decreases because the mutual interference and jamming reduce the signal-to-noise ratio. This demonstrates that a SURAP network can be more survivable than a CDMA network when there are no hidden terminals and when there is a large amount of jamming noise.

3.3.2 Blocking

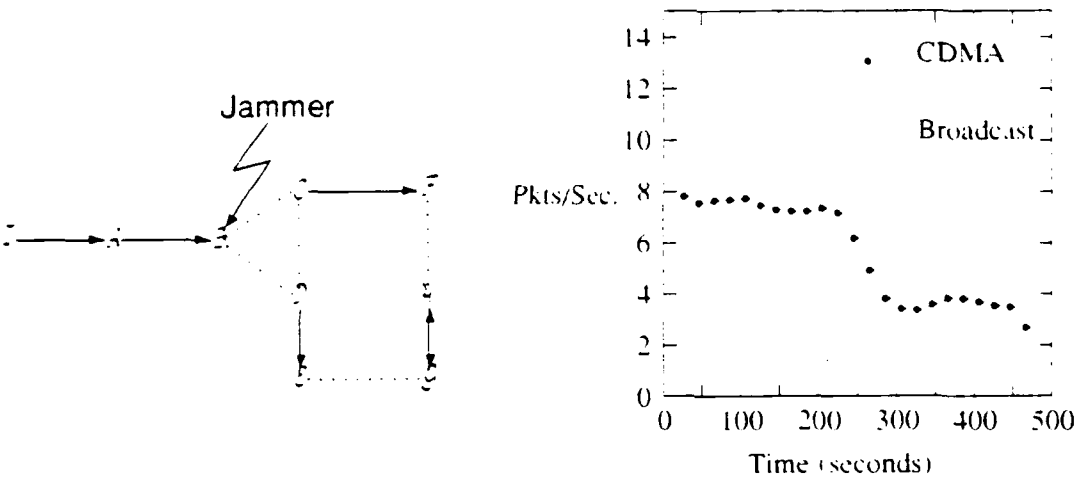


Figure 3.7: Jamming Effect on A -> B Throughput

This simulation illustrates the effect of jamming, hidden terminals, and mutual interference. Nodes E, A, and F transmit packets as fast as possible, whereas the intertransmission times of nodes D and C are independent and exponentially distributed with mean 300 milliseconds.

Note that node C and D must deal with hidden terminals. For example when node D transmits to node E it is possible that node E will not hear the packet because it is listening to node E's transmission. Therfore, node D must retransmit, causing additional blocking and mutual interference for packets transmitted from node A to node B.

Even as the jamming at node B increases the throughput of the CDMA network is greater than the SURAP network. Thus CDMA networks can be more survivable when there are many hidden terminals present.

3.4 Acknowledgement Traffic Effects

3.4.1 Mutual Interference

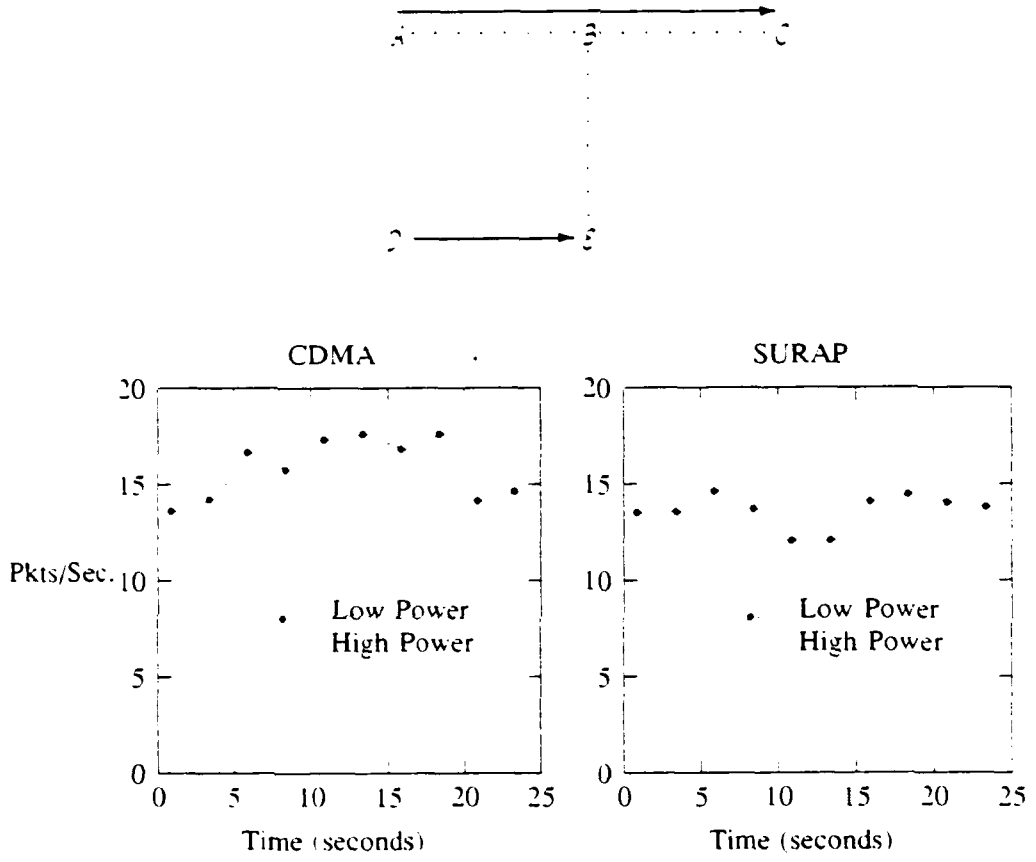


Figure 3.8: Route $\mathcal{D} \Longrightarrow \mathcal{E}$ Throughput

This simulation shows the benefit of transmitting short active acknowledgements instead of longer passive acknowledgements. The power level used by node $\mathcal{B}$ when it transmits to node $\mathcal{A}$ determines the amount of mutual itnerference these transmissions create for node $\mathcal{E}$'s receptions from node $\mathcal{D}$. During the experiment the link-gain-adjustment algorithm's setting for node $\mathcal{B}$'s transmissions to node $\mathcal{A}$ were overridden and set to either the maximum or minimum possible power level to illustrate this effect. Notice that SURAP is quite sensitive to the power used by acknowledgements; the throughput on the $\mathcal{D} \Longrightarrow \mathcal{E}$ route decreases about 25% when the acknowledgements are transmitted at maximum power. The CDMA network, on the other hand, appears to be insensitive to these power levels. The difference is due to the relative sizes of the two kinds of acknowledgements; the passive acknowledgement is the size of a data packet and consequently generates much more mutual interference than the much smaller active acknowledgement.

3.4.2 Acknowledgement Priority and Fairness

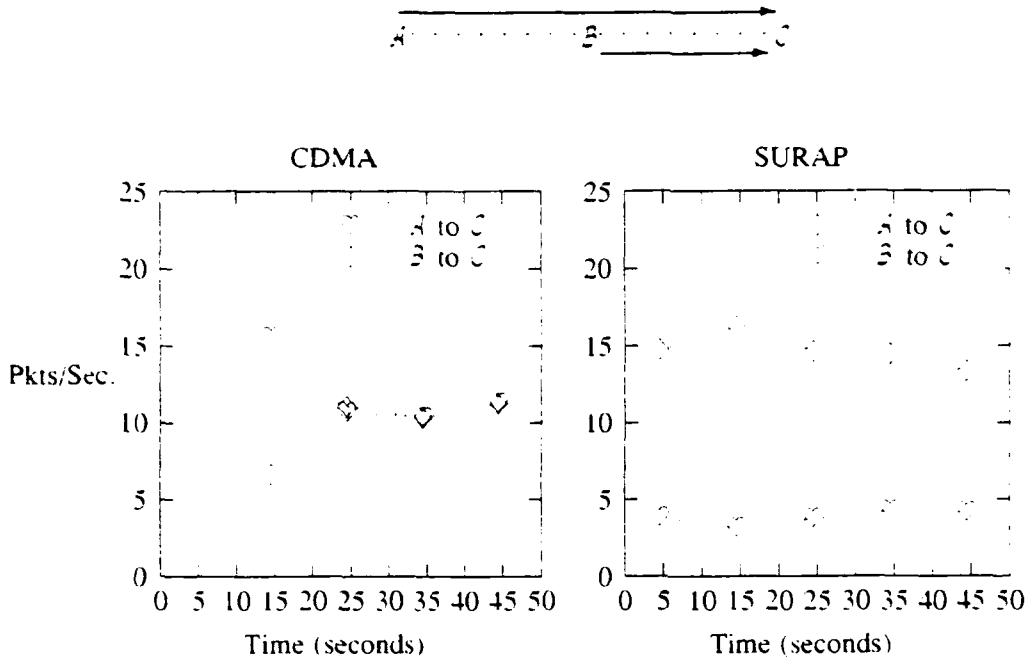


Figure 3.9: Route-by-Route Throughput

This simulation demonstrates the effect of promptly transmitting acknowledgements. In the SURAP experiment there is roughly a three-fold difference in throughput between the two routes, but in the CDMA experiment the throughputs are almost identical (after a brief warm-up period).

In the SURAP experiment node B does not acknowledge a reception from node A until it transmits the packet to node C. But node B is also the source of a route to node C, thus there is no guarantee that node B will promptly transmit any packet it has received from node A; instead it may be attempting to forward its own packet to node C. This effect causes the Pacing algorithm to slow down transmissions from node A. Because the CDMA network expedites acknowledgement transmissions this effect cannot happen.

3.5 Fairness

3.5.1 The Passing-Lane Effect

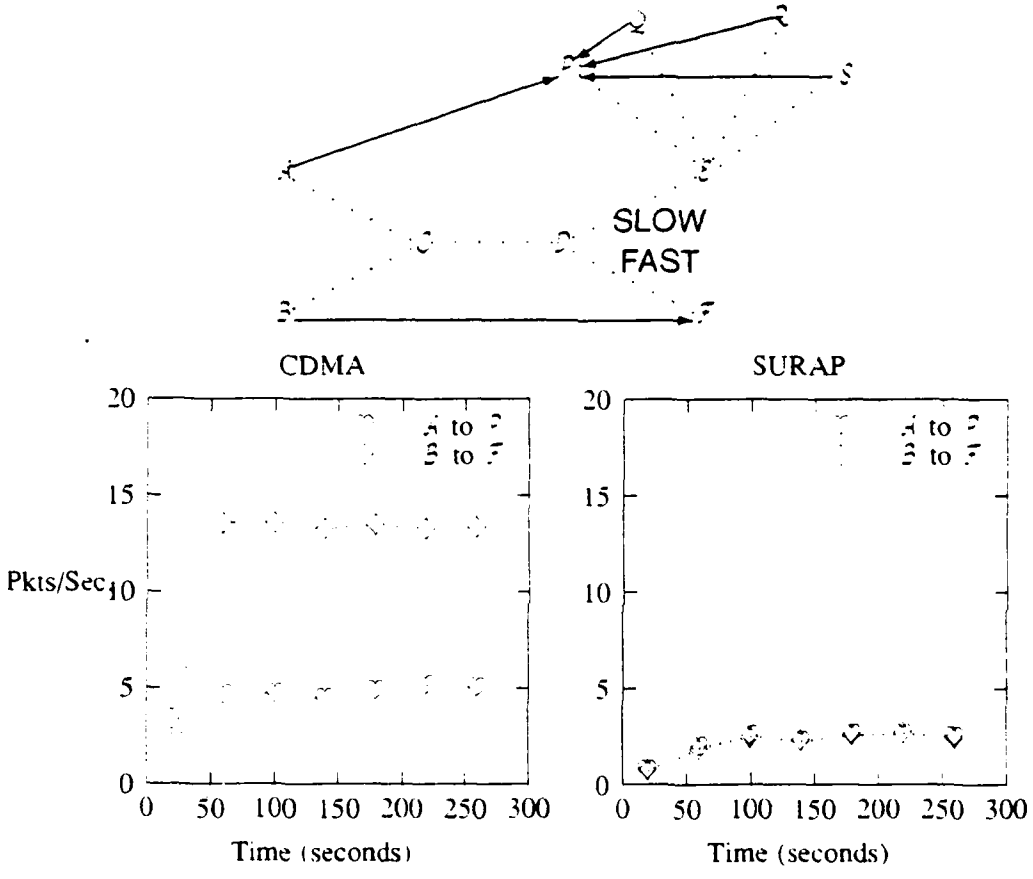


Figure 3.10: Route-by-Route Throughput

This experiment demonstrates the "passing lane" effect. Nodes A, C, D, and E all have routes to node F that must flow through node B. The heavy flow of traffic through node D makes the D-to-B link have far less capacity than the D-to-F link. When the network runs SURAP the congestion at node B spreads to the E-to-F route, but when the network runs the CDMA algorithms node D provides a passing lane for the E-to-F traffic and prevents the spread of congestion.

3.5.2 Splits and Merges

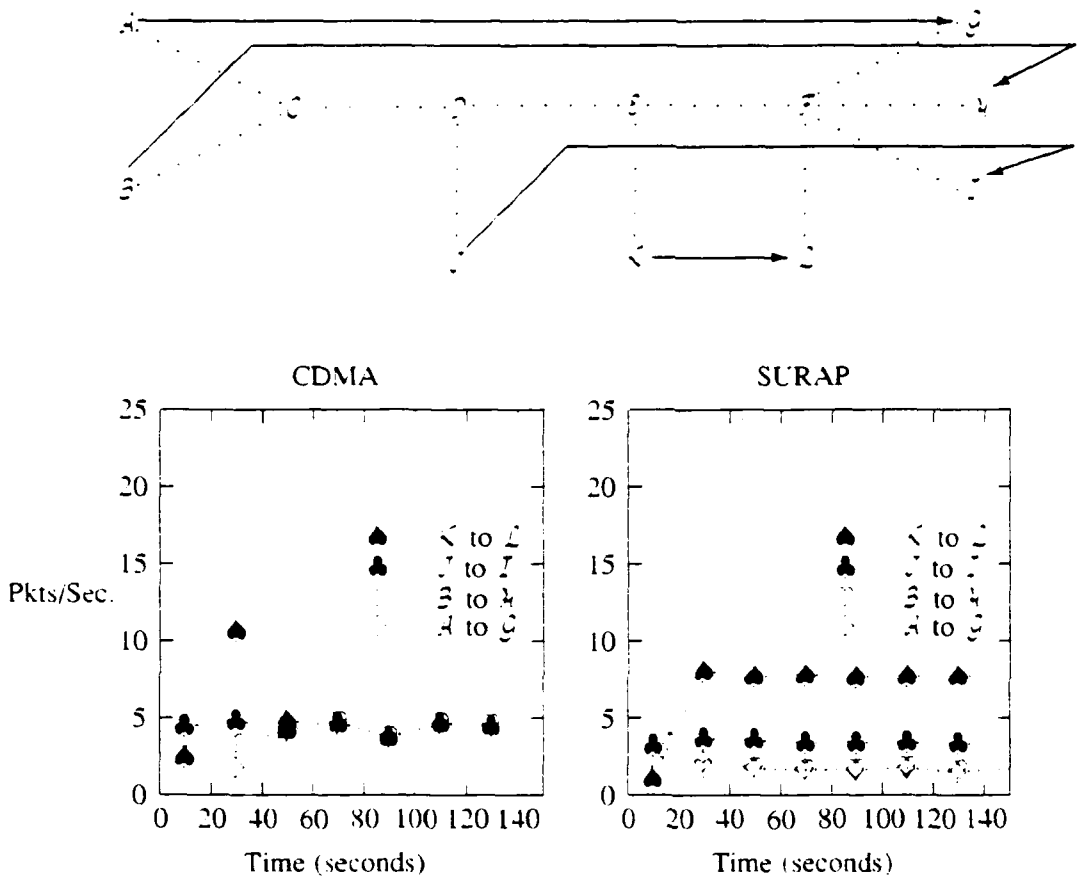


Figure 3.11: Route-by-Route Throughput

This experiment demonstrates that the end-to-end flow-control algorithm in the CDMA network assigns max/min fair end-to-end flow rates, but the SURAP network does not. For this particular topology all traffic must pass through a single node (node 5), thus max/min fairness in this case is the same as equality.

3.6 Fault Recovery

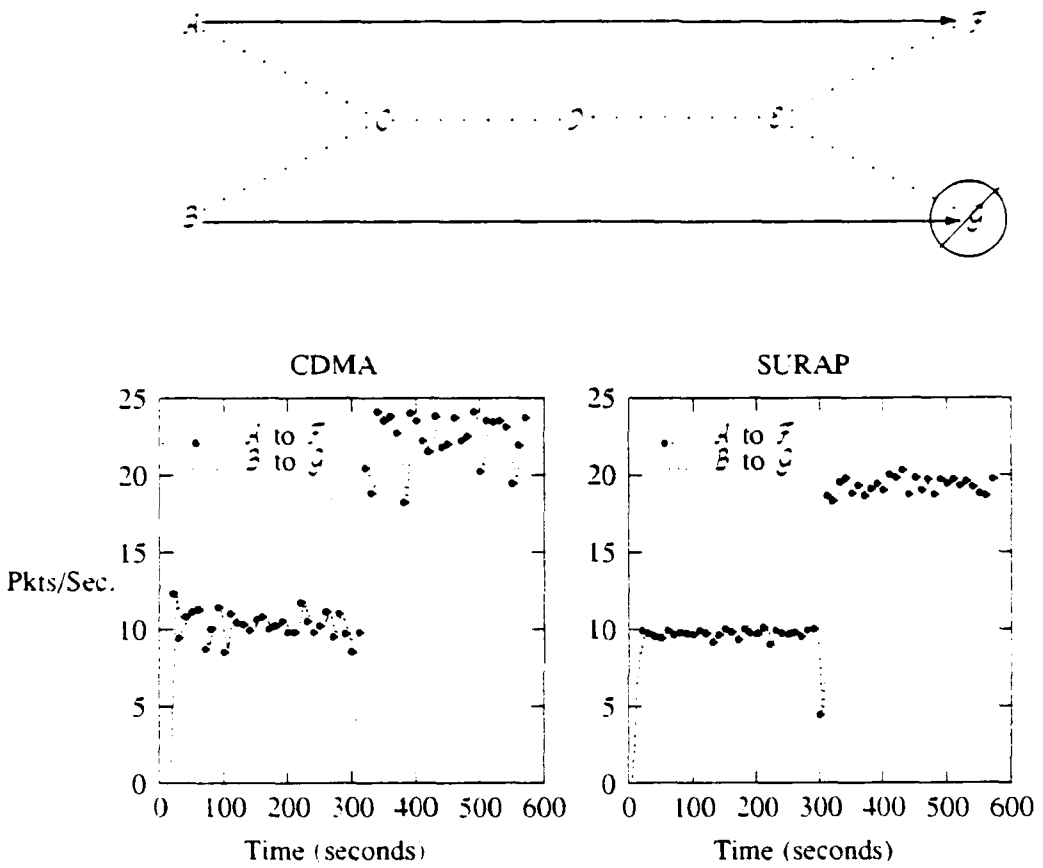


Figure 3.12: Route-by-Route Throughput

This experiment shows an example where a node failure spreads congestion when the network is running SURAP but not when when it runs the CDMA algorithms. Node G is powered down 300 seconds into the simulation. SURAP causes the A-to-F throughput to drop for a few seconds. This effect has never been detected with the CDMA network.

4. Conclusions

We have demonstrated that for many network topologies the CDMA algorithms produce greater throughput than the SURAP algorithms. Also, the CDMA algorithms provide the additional benefit of max,min-fair bandwidth allocation. But, becuase the performance of a packet-radio network is so strongly dependent on network topology, we cannot say in general that the CDMA algorithms are superior to the SURAP algorithms. Nevertheless we beleive that the CDMA algorithms are superior to the SURAP algorithms for most network topologies and when this is the case the performance difference is quite remarkable.

The most obvious limitation of the CDMA algorithms is demonstrated by Figure 3.6. This example shows that when a packet-radio network has no hidden terminals, mutual interference coupled with jamming can render the SURAP algorithms superior to the CDMA algorithms. We beleive that additional research is needed in this area. Perhaps a mechanism can be developed that will allow a packet radio to detect the difference between jamming and mutual interference and adjust its power levels and transmission rate accordingly. Perhaps this result would make CDMA algorithms superior to the SURAP algorithms for all network topologies.

Bibliography

[1] John Jubin. Current packet-radio network protocols. In *IEEE INFOCOM*, March 1985.

[2] J. Jubin and J. Tornow. The DARPA packet-radio network protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.

[3] N. Gower and J. Jubin. Congestion control using pacing in a packet-radio network. In *IEEE Milcom*, pages 23.1–1 to 23.1–6, October 1982.

[4] W. C. Fifer and F. J. Bruno. The low-cost packet radio. *Proceedings of the IEEE*, 75(1):33–42, January 1987.

[5] M. G. Lewis and J. J. Garcia-Luna-Aceves. Packet-switching applique for tactical VHF radios. In *IEEE Milcom Conference Record*, pages 449–455, Washington D. C., October 1987.

[6] J. Escobar, G. Lauer, and M. Steenstrup. *A Rate-Based Congestion-Control Algorithm for the SURAP 4 Packet Radio Architecture (SRNTN-72)*. Technical Report 7171, BBN Systems and Technologies Inc., 10 Moulton St., Cambridge MA 02138, December 1989.

[7] J. Escobar. *Radio-Parameter Selection Algorithm for Receiver-Directed Packet-Radio Networks (SRNTN-73)*. Technical Report 7172, BBN Systems and Technologies Inc., 10 Moulton St., Cambridge MA 02138, December 1989.

[8] J. R. Zavgren and G. S. Lauer. The performance improvement from receiver-directed transmissions in packet-radio networks. In *IEEE Tactical Communications Conference Record*, pages 65–72, Fort Wayne Indiana, May 1988.

[9] J. R. Zavgren. High-throughput channel-access algorithms for CDMA packet-radio networks. November 1989. In submission to IEEE Transactions on Communications.

[10] N. Shacham, E. B. Brownrigg, and C. A. Lynch. A packet-radio network for library automation. In *IEEE Milcom Conference Record*, pages 456–461, Washington D. C., October 1987.

[11] M. Leib. *Unfairness Caused by Quadratic Backoff in a Packet-Radio Network*. Technical Report 7174, BBN Systems and Technologies Inc., 10 Moulton St., Cambridge MA 02138, December 1989.

[12] P. Bausbacher and J. Kearns. Link quality estimation and transmission parameter selection in an adaptive packet-radio network. In *IEEE TCC '90*, April 1990.

Appendix A. End-to-End Flow Control

This appendix provides a brief overview of the end-to-end flow-control algorithm used by our CDMA packet radio network. We demonstrate the need for end-to-end flow control with a simple example that shows how the pacing algorithm in SURAP causes unfair end-to-end flow allocations. Consider the following example network with three routes (indicated by dotted lines):

1. $\mathcal{A} \Longrightarrow \mathcal{F}$
2. $\mathcal{B} \Longrightarrow \mathcal{F}$
3. $\mathcal{E} \Longrightarrow \mathcal{F}$.

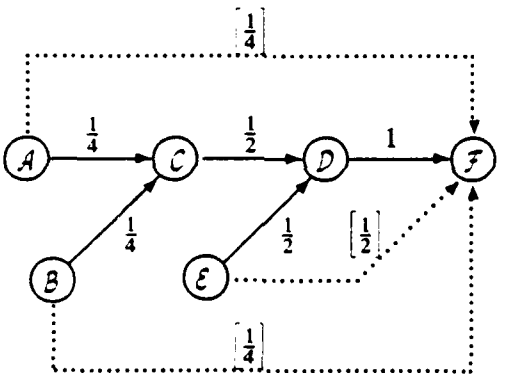


Figure A.1: Unfairness Caused by Pacing

SURAP uses no explicit end-to-end flow-control mechanism to allocate bandwidth to routes. Instead, end-to-end flow rates are an implicit function of the link flow rates set by the pacing algorithm, the single-threading policy, and fairness queueing. This design results in fair bandwidth allocations to the inbound flows at each packet-radio but it causes unfair bandwidth allocation to routes.

Nodes $\mathcal{C}$ and $\mathcal{E}$ measure node $\mathcal{D}$'s forwarding delay to node $\mathcal{F}$ to determine their pacing delay to node $\mathcal{F}$. Their individual measurements will vary somewhat but their long-term average values are identical because node $\mathcal{D}$ uses the same (long-term average) pacing delay for transmitting packets received from either node. Thus in steady state they have the same average pacing delay to node $\mathcal{D}$. Likewise, nodes $\mathcal{A}$ and $\mathcal{B}$ will have the same average pacing delay to node $\mathcal{C}$. This implicit flow allocation gives the $\mathcal{A} \Longrightarrow \mathcal{F}$ and the $\mathcal{B} \Longrightarrow \mathcal{F}$ routes half the bandwidth of the $\mathcal{E} \Longrightarrow \mathcal{F}$ route. (All bandwidth values are in packets per second in units relative to the bandwidth on the $\mathcal{D} \Longrightarrow \mathcal{F}$ link. The route bandwidths are

given in square brackets.) This network is a generic example — SURAP only provides fair end-to-end flow allocations for networks with special topologies and path losses.

The CDMA end-to-end flow control algorithm, on the other hand requires each packet radio node to compute a *path ration*, its objective maximum aggregate inbound flow rate, by monitoring transceiver utilization, CPU utilization, buffer use, and output link bandwidth. These measurements are compared to certain thresholds to determine whether the path ration should be increased or decreased. The minimum path ration along each route is communicated back to its source with a distributed algorithm that piggybacks rations in link-level control packets. (Section 2.2.1) The flow rate on each route is then throttled according to the minimum path ration. Note that each node computes a single path ration that effects all routes equally. In this way all route sources sharing a bottle neck are throttled equally while those route sources not affected by the bottleneck are not throttled. The algorithm recovers from node and link failures by throttling affected traffic until the problem is corrected or routes are changed.

A performance study[6] shows the algorithm provides max/min fair flow control except for certain unusual situations where channel contention creates dependence among neighboring nodes. Comparison of the network flows with and without the algorithm shows that it maintains acceptable delay without significantly restricting the aggregate network throughput.

One of the major strengths of the algorithm is that it can achieve max/min fair end-to-end flow allocation. This means that only sources affected by a bottleneck are restricted and these sources are restricted in a fair manner. This is particularly important for large packet-radio networks where unfairness could lead to extremely poor throughput for certain users even though average network throughput is high.

Appendix B. Adaptive Link Gain Adjustment

This algorithm dynamically sets the bit rate, power level, and FEC rate on a packet-by-packet basis. These parameter settings impact the signal-to-noise ratio and signal-processing gain at each LPR in radio range; the combined effect at a particular LPR is called the link gain. The link gains have a profound impact on the success of packet transmissions and thus network performance. Link gain is increased by increasing the energy per transmitted bit, by using higher power, a lower bit rate, or "stronger coding" (lower FEC rate). While increasing the gain on any link will always make it more reliable and increase its throughput, a unilateral gain increase can actually decrease the reliability and throughput of adjacent links by increasing mutual interference. The Adaptive Link Gain Adjustment Algorithm lets LPRs cooperatively adjust their link gains to make link throughput compromises that increase the network throughput. SURAP[12] and the CDMA link-layer[7] use slightly different versions of the same basic algorithm. These differences do not have a major impact on our performance results.

Appendix C. LPR Hardware Description

C.1 Channel Access

Each LPR node contains a half-duplex radio transceiver that is always in one of three mutually exclusive states: monitoring, receiving, or transmitting. Each packet transmission begins with a short preamble that establishes bit-synchronization and designates a pseudo-noise spreading sequence for receiving the packet. When an LPR begins transmitting, every LPR in radio range that is monitoring the channel receives the preamble and establishes bit and frame synchronization. The preamble can designate one of four spreading sequences: a common broadcast sequence, either of two "group" codes, or a special receiver-directed sequence which is different for each LPR (and is a deterministic function of its ID).

When a preamble is decoded, the receiver attempts to de-spread the rest of the packet. If the sequence it is using doesn't match the one used by the transmitter, the LPR loses synchronization and resumes channel monitoring or starts a transmission. Otherwise it receives the packet and stores it in a buffer. In the case of receiver-directed transmissions, each LPR receiving a packet will try to use its own unique sequence to despread the signal — only the intended receiver can succeed, leaving the remaining LPRs free to receive packets destined to them after they quickly lose synchronization.

An LPR using the CDMA protocols has the capability of either broadcasting or receiver-directing its transmissions on a packet-by-packet basis. The broadcast mode is used for distributing routing information and for network entry. Data packets are normally transmitted with the receiver-directed codes. But in exceptional circumstances, e.g., link failures, a packet can be broadcast so any neighbor closer to the destination can forward it, a procedure known as alternate routing that is explained in Section (2.2.3).

C.2 Link Gain Settings

The LPR can control its transmitter's bit rate (100 or 400 Kbps), FEC rate (1:1, 7:8, 3:4, or 1:2), and power level (five watts with either none, 8, 16, or 24 dB of attenuation) on a packet-by-packet basis. These 24 link-gain settings, coupled with a 128 chip-per-bit spreading factor give the LPR the capability of counteracting signal attenuation, mutual interference, multi-path reflections, and jammers. The Adaptive Link Gain Adjustment Algorithm (Section (B)) implements a feedback loop that quickly changes these settings in response to environmental changes.